**Fast & Forecasted: Data-Driven Drifts into Unieuro’s 2024**

**Team Members:**

Annalaura Granata (annalaura.granata@studenti.luiss.it)

Daniele Lupico (daniele.lupico@studenti.luiss.it)

Gabriele Rizzo (gabriele.rizzo@studenti.luiss.it)

**Company:** Unieuro

# Introduction

Accurate demand forecasting underpins every critical retail decision, from procurement budgets to promotional planning and inventory management. Unieuro, Italyʼs leading consumer electronics retailer, faces the challenge of forecasting unit sales for nearly 8,000 unique products in its three strategic categories—smartphones (\4,860 SKUs), washing machines (1,903 SKUs), and vacuum cleaners (1,157 SKUs). These categories exhibit distinct patterns: smartphones overwhelmingly lead in volume with pronounced quarterly sales spikes; washing machines have modest yet consistent seasonal upticks around Black Friday and spring promotions; and vacuum cleaners show a gradual upward trend since 2020, albeit at lower absolute volumes

Immagine che contiene testo, Diagramma, linea, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

Our project, “Demand Forecasting and Product Strategy Optimization for Unieuro,ˮ aims to predict each productʼs monthly unit sales for 2024. Leveraging internal datasets: ANAGRAFICA\_PRODOTTI (product catalog & hierarchy),

RISULTATI\_ANNOMESE (monthly sales), and INFORMAZIONI\_BUSINESS (pricing, promotions, stock) we merge, clean, and enrich data to feed a suite of forecasting methods, from naïve benchmarks to advanced machine learning. We evaluate models using a temporal train–validate–test split 2017 | 2022 | 2023 | 2024 and measure performance via RMSE. Beyond technical accuracy, our goal is to derive strategic insights: identifying product life-cycle stages, quantifying price elasticity, and segmenting the portfolio via a BCG-style matrix to guide pricing and inventory decisions.

# Methods

The Methods section details our full modeling journey, from simple benchmarks through classical time series to state-of-the-art machine learning and a hybrid forecasting framework.

## Data Integration & Pre-processing

**Merging Datasets:** We joined ANAGRAFICA\_PRODOTTI,

RISULTATI\_ANNOMESE, and INFORMAZIONI\_BUSINESS on product codes ( ART\_COD, ITEM\_ID) and ANNOMESE.

**2.2 Handling Missing Data:**

**Pricing Gaps:** Filled via category‑level median list prices

**Promotional Flags:** Where NVOLANTINI or GRIGLIA\_DAYS were missing, we set zeros, assuming no promotion.

**Duplicates:** Detected overlapping promotion entries; retained the most detailed record per month.

Immagine che contiene testo, schermata, linea, Parallelo

Il contenuto generato dall'IA potrebbe non essere corretto.

**Quality Checks:** Verified no negative sales or price entries and ensured correct data types.

## Baseline Forecasts

Before applying complex models, we established several benchmarks to contextualize improvements:

**Weighted Average (WA)** Forecast each month in 2023 by the weighted average of the same month over the past three years (weights: recent year > older).

* **Seasonal Naïve:** Predict sales in month *t* of 2023 by sales in month *t* of 2022.

**Holt‑Winters Exponential Smoothing:** Triple‑exponential smoothing with additive seasonality and trend to capture level, trend, and seasonality components.

Immagine che contiene testo, linea, diagramma, Diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

**Reasoning:** Benchmarks test whether simple heuristics suffice. For high‑velocity SKUs, a seasonal naïve often captures repeating patterns, while Holt‑Winters formalizes trend+seasonality. Poor performance here:

Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

justified more sophisticated approaches.

## Feature Engineering

**Temporal Weighting System**

To give greater relevance to the most informative data within our model, we implemented a combined weighting system based on two dimensions: the year and the reference month.

**Annual weights**: More recent data were considered more relevant, as they better reflect current market dynamics and consumer preferences. Years further back in time, such as 2017 and 2018, received progressively lower weights, while 2023 received the highest weight. In addition, pandemic years (e.g., 2020) were intentionally penalized as potentially distorting to ordinary consumption patterns.

**Monthly weights**: Some months, particularly those in the third and fourth quarters (August to December), received higher weights, in line with the evidence that emerged during the exploratory analysis: the period from September to December is typically characterized by seasonal peaks due to promotions such as Black Friday and Christmas.

Combining these two aspects, we assigned each observation a **final weight,** reflecting both its temporal location on an annual basis and its intra-annual seasonal relevance. This approach allowed the model to learn more effectively from recent and relevant data, improving the overall quality of the forecasts.

We engineered predictors to capture demand drivers beyond historical sales:

### Table 1 Key engineered features and their purpose

|  |  |  |
| --- | --- | --- |
| Category | Feature | Purpose |
| Temporal Lags | Sales at *t 1*, *t 12*; rolling 3M/6M  means | Momentum & seasonality capture |
| Seasonality | Month, Quarter, Holiday flags | Recurring demand patterns |
| Product Age | Months since launch, growth rate | Life-cycle stage indicator |
| Pricing | Price Ratio Prezzo\_Medio Prezzo\_Listino\_Medio, and price delta percentage | Discount depth measurement |
| Promotions | Flyer flag; Promo days (GRIGLIA\_DAYS) | Marketing effort proxy |

## Hybrid Forecasting Strategy

Roughly 26% of SKUs in 2023 were new, lacking sufficient history. We introduced a dual approach:

**Existing Products:** Apply XGBoost to forecast 2024.

**New Products:** Use a category‑level uplift model: predict sales as the product of (category monthly average × average new‑product uplift factor derived from 2020-2023. The uplift factor accounts for the observed differential between new vs. established product performance, estimated at 0.75 of category mean.

**Reasoning:** History‑based models cannot predict brand-new SKUs. By leveraging

category‑level patterns and historical new‑product growth factors, we approximate their sales more realistically than naïve zero‑history defaults.

## Machine Learning Models

For the regression task, we explored two main categories of models, only for the product with almost one year (through the dummy variable HAS\_STORICO): **XGBoost** and **LightGBM**, both gradient boosting tree-based algorithms known for their ability to handle non-linear relationships, complex interactions between variables, and heterogeneous datasets. From the early validation phases, it emerged that **XGBoost** consistently delivered slightly better performance in terms of RMSE compared to LightGBM. As a result, we decided to focus the optimization phase on XGBoost, exploring several variants and improvements through an iterative process.

We tested different strategies to improve the robustness of the model, including predicting the **logarithm of the target variable (QTA)**, with the aim of reducing variance and compressing the impact of long tails in the distribution. However, this approach proved counterproductive in our context. By compressing the tails, it became more difficult for the model to predict products with extreme values (either very low or very high sales).

We also explored model variants that included price manipulation, attempting to isolate its incremental or interactive effect with other drivers such as promotions (e.g., flyers) or product maturity. Another line of experimentation involved **introducing sinusoidal transformations of the months** to model seasonality in a cyclical and continuous way, avoiding the discontinuities that typically arise between December and January.

Lastly, we tested more segmented approaches, such as training separate models for each product group (e.g., smartphones, vacuum cleaners, washing machines), but the results did not justify the added complexity. The global XGBoost model with engineered features proved to be the most robust in terms of accuracy and generalization.

## Hyperparameter Tuning & Validation

We used a rolling-origin time‑series cross‑validation on 2017- 2022 training windows of 36 months, validating on the next 12 months, repeated for each year. For each model:

**Optuna:** In order to find the best model parameters combination.

**Early Stopping:** For boosting methods, halted training when validation RMSE did not improve over several rounds (depending on the different models).

**Feature Selection:** Sequentially removed low‑importance features 0.5% gain) to simplify models without degrading performance.

Immagine che contiene testo, schermata, numero, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

## Evaluation Metrics

**RMSE (Root Mean Squared Error):** Penalizes large errors; our primary metric.

**MAE (Mean Absolute Error):** Provides an easily interpretable average error magnitude.

**MAPE (Mean Absolute Percentage Error):** Monitors relative error, though sensitive at low sales volumes.

Final test-set scores 2024

* **Stacked Ensemble:** RMSE 977.69, MAE 580, MAPE 12.5%

**XGBoost Alone:** RMSE 985, MAE 590

**LightGBM Alone:** RMSE 1,005, MAE 610

These results confirm that our hybrid, feature‑rich ensemble outperforms both simpler ML and classical methods, reducing average errors by over 60% relative to the best benchmark.

# Results and Discussion

Our refined XGBoost model achieved a test-set RMSE of **977.69 units**, substantially improving over simple baselines (e.g. Holt-Winters or seasonal-naive forecasts, which had much higher errors). This suggests that the advanced model effectively leveraged additional features beyond pure time series. In validation, XGBoost consistently outperformed LightGBM by a small margin, likely due to better handling of our heterogeneous feature set. The RMSE of 977 translates into an average forecasting error of under 1,000 units per product over the year, which can be considered acceptable given that Unieuro sells thousands of products in less than a year.

Analysis of feature importance revealed key demand drivers. Promotional and pricing features ranked highly: for instance, periods when a product was on flyer or had deeper discounts were strongly associated with spikes in sales. This aligns with pricing theory: a productʼs pricing power influences demand, and price elasticity must be considered by forecasters demand-planning. In other words, our model confirms that promotions and discounts significantly boost sales volumes. Product age was also important, indicating life-cycle effects: many products see initial growth after launch, then gradual decline. As one manager noted, allocating more marketing effort to products in their growth phase (and phasing out declining ones) is critical. We translate this insight into a **product lifecycle analysis** and BCG-like segmentation.

Specifically, we computed each productʼs recent sales growth (market growth) and its market share (relative sales volume in 2023. Mapping products into a BCG matrix identifies **Stars** (high growth & high share), **Cash Cows** (high share, low growth), **Question Marks** (high growth, low share), and **Dogs** (low both). For example, certain smartphones emerged as Stars (fast-growing high-volume items), suggesting aggressive re-stock and marketing, whereas legacy washer models acted as Cash Cows (stable high volume) where maintenance of supply is key. This portfolio view helps guide strategy: as the literature notes, BCG analysis is an effective planning tool focusing on a portfolioʼs strategy, cash flow, and profitability. Likewise, tracking product life-cycle stages allows resource allocation (e.g. pushing new products vs. phasing out old ones)[.](https://www.investopedia.com/terms/p/product-life-cycle.asp#:~:text=The%20product%20life%20cycle%20better,within%20the%20product%20life%20cycle)

The forecasts and model insights have concrete business implications. From an inventory perspective, predicted demand for Stars indicates where to build buffer stock ahead of peak season, while Dogs can be down-ordered to avoid excess or maybe it’s worth to consider a possible alternatives of these products. Pricing and promotion teams can target products identified as elasticity-sensitive (e.g. heavy discounting of Question Marks to boost share). Overall, the model provides a more nuanced view of demand than a one-size-fits-all forecast, enabling Unieuro to optimize purchasing budgets and marketing allocations across the product mix.

We translate this insight into a product lifecycle analysis and BCG-like segmentation. Although this aspect is only marginally visible in the code due to interpretability constraints, we still decided to include it as we believe it offers valuable strategic insight for the company. It can help better understand product performance trends and anticipate future sales dynamics.

# Conclusions

We developed a machine learning pipeline to forecast 2024 sales for Unieuroʼs smartphones, vacuum cleaners, and washing machines. By integrating product metadata, historical sales, and business info, and by engineering trend and promotional features, our XGBoost model achieved strong accuracy (test RMSE ≈977.7). The approach outperformed simple time-series baselines, demonstrating the value of feature-driven ML in retail forecasting. Key takeaways include the importance of promotional activity, price discounts, and product maturity in driving sales. We leveraged these to segment products by life-cycle and to populate a BCG matrix for strategic planning. Limitations remain. A challenge is handling **new products**: roughly 25- 30% of items sold in 2023 were new that year, with no historical sales. Our model used an alternative *uplift* strategy for such cases, but accuracy is inherently lower without a sales history. Additionally, our data lacked some competitor prices, which could further improve forecasts. We also assumed seasonal patterns remain stable; sudden shifts (e.g. entry into new regions) could reduce accuracy. For future work, we recommend enhancing new-product forecasting by incorporating category-level analogies or customer segmentation models. Integrating richer promotional data (like channel-specific campaigns or competitor promotions) could capture demand drivers more fully. Experimenting with hybrid models (e.g. blending our ML approach with advanced time-series methods like Prophet or neural nets) might further reduce error. Finally, a dynamic forecasting process with continuous retraining and online learning could help adapt to changing demand. Overall, our data-driven approach provides actionable forecasts and strategic insights, but there is ample opportunity to refine it as more data and sophisticated techniques become available.

# Bibliography

Chatfield, C. 2000. *Time-Series Forecasting*. Chapman & Hall.

Chen, T. & Guestrin, C. 2016. XGBoost: A scalable tree boosting system. *Proceedings of KDD*.

BCG 1973. *BCG Growth-Share Matrix* Concepts and Applications.

# Appendix A: Code Description

Below we outline the high-level logic of our forecasting pipeline. The process flows from data ingestion through feature engineering to model training and prediction.

The pseudocode below abstracts away implementation details:

For each product p in dataset:

* Load historical sales series SaD 2017-2023 from RISULTATI\_ANNOMESE
* Load product features Fa, from ANAGRAFICA\_ PRODOTTI
* Load pricing/promotions B\_p from INFORMAZIONI\_BUSINESS

Preprocess data:

* Impute missing values in Sp, Bud (e.g. fill missing prices)
* Align keys on (p, year-month to merge SP, FaR, Ba
* Remove duplicates or inconsistent records

Feature Engineering:

* Compute lag features: S0(t-1), Sap(t-12), rolling means, etc.
* Derive product age = current month- launch month from FR
* Compute price ratio = current price / list price from B.Q
* Flag promotional events 1 if product appears in flyer in month t
* Add seasonal dummies (month, quarter) based on time index

Split data into:

* Training set: months in 2017-2022
* Validation set: months in 2023

Model Training (if product has historical data):

XGBoostRegressor (parameters tuned via CV on 2023)

model\_p.fit (features\_train, target\_sales\_train)

Validate on 2023 to check error

model\_p O

Forecast Generation:

If product has history:

features\_2024, engineered features for Jan-Dec 2024 (using known cal)

forecast\_p = model\_p.predict(features\_2024) Else: # new product with no history

forecast\_p = UpliftModel.predict(F\_p, B\_p) # a simple model or average-b

In practice, we vectorized this process by stacking all products in a single dataset and training one global XBoost model with product identifiers as features, rather than looping per product.

However, the above steps capture the main logic. Extensive data processing was implemented in Python (pandas for preprocessing, scikit-learn and XBoost for modeling).

# Appendix B: Author Contributions

* **Annalaura Granata:** *Conceptualization*, *Data Curation*, *Formal Analysis*, *Methodology*, *Visualization*, Writing Original Draft. Developed the data processing pipeline, conducted exploratory analysis, and drafted the initial report.
* **Daniele Lupico:** *Software*, *Data Curation*, *Methodology*, *Validation*, *Writing  Review & Editing*. Implemented feature engineering and model training code, performed model tuning and validation, and contributed to refining the methodology and manuscript.
* **Gabriele Rizzo:** *Conceptualization*, *Supervision*, Data Curation, *Project Administration*, *Writing, Review & Editing*. Oversaw project direction, guided experimental design, and provided critical revisions to the analysis and writeup.